A Comparative Study on Vehicle Physical Appearance Identification using Transfer Learning Methods

Kahabodee Prakobchat[1], Kwankamon Dittakan[2,*], and Salang Musikasuwan[3]

[1]College of Digital Science, Prince of Songkla University,
15 Karnjanavanich Rd., Hat Yai, Songkhla 90110, Thailand
[2]College of Computing, Prince of Songkla University, Phuket Campus
80 M.1 Vichitsongkram Road, Kathu, Phuket 83120, Thailand
[3]Faculty of Science and Technology, Prince of Songkla University, Pattani Campus
181 Charoenpradit Road, Rusamilae, Muang, Pattani 94000, Thailand

6410025005@psu.ac.th, kwankamon.d@phuket.psu.ac.th*, and salang.m@psu.ac.th

Abstract. *Traffic rule violations by drivers are a significant global concern, particularly in urban areas, as they contribute to increasing traffic accidents. This study proposed a novel approach to identifying vehicles involved in such violations by building prediction models using image processing and machine learning techniques. The research focused on three key vehicle characteristics: type, colour, and brand. The study employed a transfer learning mechanism as the machine learning method to generate the prediction models. The results revealed that the YOLO V8 achieved the highest accuracy in predicting vehicle type and colour, with an accuracy of 98.9% and 93.2%, respectively. Comparatively, YOLO V7, V6, V5, V4, and V3 achieved lower accuracies. In terms of predicting vehicle brand, the YOLO V8 achieved an accuracy of 89.8%, surpassing the accuracies of the YOLO V7, V6, V5, V4, and V3. These findings demonstrated the potential of image processing and machine learning techniques in accurately identifying vehicles involved in traffic violations and highlighted the opportunity to develop effective strategies to reduce the number of traffic accidents caused by rule violations. This research has significant implications for enhancing road safety and promoting advanced technologies to address real-world problems.*

Received by	03 July 2023
Revised by	20 September 2023
Accepted by	03 October 2023

Keywords:

Vehicle appearance detection, deep learning, object detection, transfer learning, convolutional neural network

1. Introduction

Vehicle detection is crucial for many traffic surveillance applications to ensure fairness and credibility. In Thailand, where a high population and vehicle density can lead to traffic violations, road accidents, vehicle theft, and public safety concerns, vehicle detection becomes even more critical. However, traffic police and government officers often rely on eyewitness accounts to gather information on lost vehicles, leading to investigation delays and inaccurate details. Also, Thailand has the ninth highest rate of accidents in the world, with approximately 22,491 deaths per year, which is two times higher than the global average, according to a report by the World Health Organization (WHO) in 2018 [1]. The Office of Transport and Traffic Policy and Planning in the Ministry of Transport has identified three leading causes of accidents in Thailand, one of which is people's actions, such as speeding or breaking traffic laws.

Vehicle Physical Appearance Identification is important for several reasons. Accurately identifying vehicles based on their physical characteristics is crucial for law enforcement agencies and surveillance systems. It helps officers verify and track vehicles involved in criminal activities such as theft, hit-and-run incidents, or other illegal actions. The ability to quickly and reliably identify vehicles enables law enforcement to efficiently investigate and take appropriate action.

Identifying vehicles based on their external appearance contributes to improved traffic management and safety. It allows for the detection of traffic rule violations, such as speeding, reckless driving, or running red lights. By identifying vehicles involved in such violations promptly, appropriate measures can be taken to enhance traffic safety, such as issuing warnings or citations to drivers or implementing targeted enforcement strategies.

Vehicle physical appearance identification aids in accident investigations by providing crucial information about the vehicles involved. Determining the type, colour, brand, and model of vehicles helps reconstruct the sequence of events leading to an accident. This information is valuable for insurance claims, legal proceedings, and implementing preventive measures to reduce similar accidents in the future. Vehicle identification based on physical appearance assists in the recovery of stolen vehicles. By accurately identifying the characteristics of a stolen vehicle, law enforcement agencies can quickly locate and recover it. This contributes to reducing vehicle theft rates and increasing public safety.

Moreover, vehicle physical appearance identification can be used for traffic data analysis, providing in-depth insights into vehicle demographics, trends, and patterns. This information helps transportation authorities and urban planners make informed decisions regarding road infrastructure development, traffic flow optimization, and targeted transportation policies.

Machine learning is a computer technique that focuses on creating and developing mathematical models capable of learning and processing data automatically using input data to predict outcomes or analyze data. These models can be improved and fine-tuned through learning from existing data.

Object detection is an interesting field of application for machine learning, especially when dealing with images or videos. It aims to detect and identify the precise location of objects in images or videos. The objects that can be detected can be various, such as cars, humans, animals, and other objects. To perform object detection, it is necessary to use suitable machine learning techniques and models, such as Convolutional Neural Networks (CNN), which are popular deep learning models for object detection. CNNs are effective in handling images or videos by learning from data to recognize and identify objects. The process of performing object detection using machine learning and CNN involves various steps, including data pre-processing, building and training CNN models, object detection, and evaluating and improving the performance of the object detection system. Object detection has applications in various fields and industries, such as face detection and recognition systems in access control, security surveillance systems, traffic control systems, and research related to photography and videos. Object detection helps automate the processing of images or videos with large amounts of data, reducing the time and labour required for traditional object detection and identification.

The paper introduces a framework based on convolutional neural networks (CNNs), which are commonly used for object detection and image processing. This framework utilizes transfer learning to analyze real-time video data captured by CCTV cameras and extract specific information about vehicles, such as their type, colour, brand, and model. The proposed research has several benefits. Firstly, it enhances vehicle detection accuracy, providing more reliable results. Secondly, it enables the identification of unsafe vehicles, allowing traffic officials or drivers to be alerted promptly. By identifying potential safety hazards, this system can help reduce the occurrence of car accidents. Additionally, it has the potential to contribute to the prevention of motor vehicle theft. In summary, the framework described in the paper leverages CNNs and transfer learning to improve vehicle detection, enhance safety measures by alerting authorities and drivers about unsafe vehicles, and potentially reduce car accidents and instances of motor vehicle theft.

The remainder of this paper is structured as follows: Section 2 presents related and previous works. Section 3 introduces the proposed framework's schematic. Section 4 provides details about the dataset used in this research. Section 5 summarizes the pre-processing process. Section 6 discusses the object detection method applied in this research. Section 7 presents the experiment’s results. Finally, Section 8 concludes the paper with a discussion and summary of the work.

2. Previous Work

Object detection is a widely used technique that employs algorithms resembling human neural networks, particularly Convolutional Neural Networks (CNNs). CNNs are bio-inspired neural networks and a form of deep learning that specializes in image recognition and processing. They mimic human vision and leverage deep learning to perform both creative and descriptive tasks. In image processing, CNNs can be combined with expert systems and natural language processing, enabling them to tackle a wide range of perceptual tasks.

The CNN model is constructed using layers, starting with the input layer, which receives the input image data and normalizes it to be zero-cantered. This layer also scales the image to a range between 0 and 1, facilitating faster data training. The first layer in the CNN model is the Convolution Layer (ConvLayer), which extracts essential features from the image while preserving the spatial relationships among neighbouring pixels [2]. Unlike a regular neural network that connects all neurons from the previous layer, the ConvLayer selectively connects the region it needs to detect the desired feature, known as the Receptive Field. The filter or kernel within a ConvLayer helps extract attributes necessary for object recognition.

Typically, one filter or kernel in a CNN is responsible for extracting one specific feature of interest, necessitating multiple filters to capture various spatial attributes. Stride is employed to move the filter, determining the number of pixels by which the filter is shifted. Point products are computed between input pixels and filter weights. The convolution process, along with padding, can reduce the size of the resulting matrix. To counter this, zero-padding expands the input borders from all sides. Another technique used to downsize the feature map is pooling, often referred to as down sampling. Max-pooling is one of the commonly used pooling methods. It involves using a virtual aggregation filter, typically of size 2x2, which scans the feature map and selects the maximum value. The final stage of a CNN is the fully connected (FC) layer, where each neuron in the layer is connected to every neuron in the previous layer. These layers produce the output of the CNN [3].

Transfer learning has proven to be highly effective in object detection. Instead of training a deep learning model from scratch, which can be computationally expensive and time-consuming, transfer learning allows us to use pre-trained models that have been trained on large-scale datasets like ImageNet. These pre-trained models have learned general features and patterns that are useful for many vision tasks.

In object detection with transfer learning, the pre-trained model serves as a feature extractor. The early layers of the model capture low-level visual features like edges and textures, while the deeper layers learn higher-level features that are more specific to the dataset they were trained on. By using the pre-trained model as a feature extractor, we can extract meaningful features from input images and feed them into a separate object detection algorithm, such as a region proposal network (RPN) or a region-based convolutional neural network (R-CNN).

Previous research in vehicle detection using object detection techniques has extensively employed deep learning methods, particularly algorithms like YOLO V2, YOLO V3, YOLO V4, and YOLO V5. Several studies have demonstrated the effectiveness of these algorithms in different datasets and applications. Saribacs utilized the YOLO V2 algorithm to detect vehicles in an image dataset captured by quadcopters [3], achieving an accuracy of over 80%. Similarly, Xu applied the YOLO V3 algorithm to satellite images from the Utah AGRC dataset [4], successfully detecting aerial vehicles with high accuracy. Machiraju proposed the use of a YOLO V3 network with transfer learning for object detection [5], specifically for effectively detecting and separating occluded objects. This approach demonstrated promising results. Other research studies have explored the use of deep learning techniques beyond the YOLO algorithms. Chen employed both SSD and YOLO V3 for object detection, tracking, and distance estimation in intelligent mobility applications [6]. Corovic implemented YOLO V3 for real-time detection of traffic participants, achieving high accuracy in detecting cars, trucks, pedestrians, traffic signs, and lights in various weather and lighting conditions [7]. The works of Zhang [8], Supreeth [9], and Yang [10] also contribute to the advancements in vehicle detection using deep learning. Zhang used YOLO V2 to detect and position traffic lights based on the HSV colour model and colour ratio design. Supreeth proposed a method combining a Gaussian mixture model and transfer learning to detect objects in video frames. Yang developed a system using the SSD algorithm to classify and locate vehicles, considering aspects such as image collection, calibration, model training, and detection performance evaluation.

Overall, deep learning methods, including YOLO variants, have demonstrated effectiveness in vehicle detection tasks across various datasets and scenarios. These approaches, coupled with appropriate pre-processing techniques, contribute to advancements in object detection for vehicle-related applications.

3. Proposed Framework

The research methodology employed in this study aims to develop a framework for generating deep learning models for vehicle physical appearance identification. The following steps were undertaken to accomplish this objective: (i) Data Collection: CCTV cameras installed at an intersection road in Phuket were used to collect video footage of vehicles. The recorded data provided a comprehensive dataset for analysis and model training. (ii) Frame Extraction: The collected video footage was processed through frame extraction, converting it into a series of individual images. This step allowed for the analysis and identification of vehicles' physical appearance at the frame level. (iii) Deep Learning Model Generation: Three deep learning models were developed to identify specific aspects of the vehicle's physical appearance: Vehicle Type Detection: A model was trained to detect and classify vehicle types, including Cars, Buses, Trucks, and Motorcycles. Vehicle Colour Classification: Another model was generated to classify vehicles based on their colours. The study considered ten colours, including white, black, grey/silver/bronze, blue, red, yellow, green, blue, orange, and others. Vehicle Brand Detection: A model was developed to detect and identify various vehicle brands, such as Toyota, Isuzu, Ford, Mazda, BMW, Mitsubishi, Honda, Suzuki, Nissan, and others. Optionally, vehicle model detection could also be included in the study.

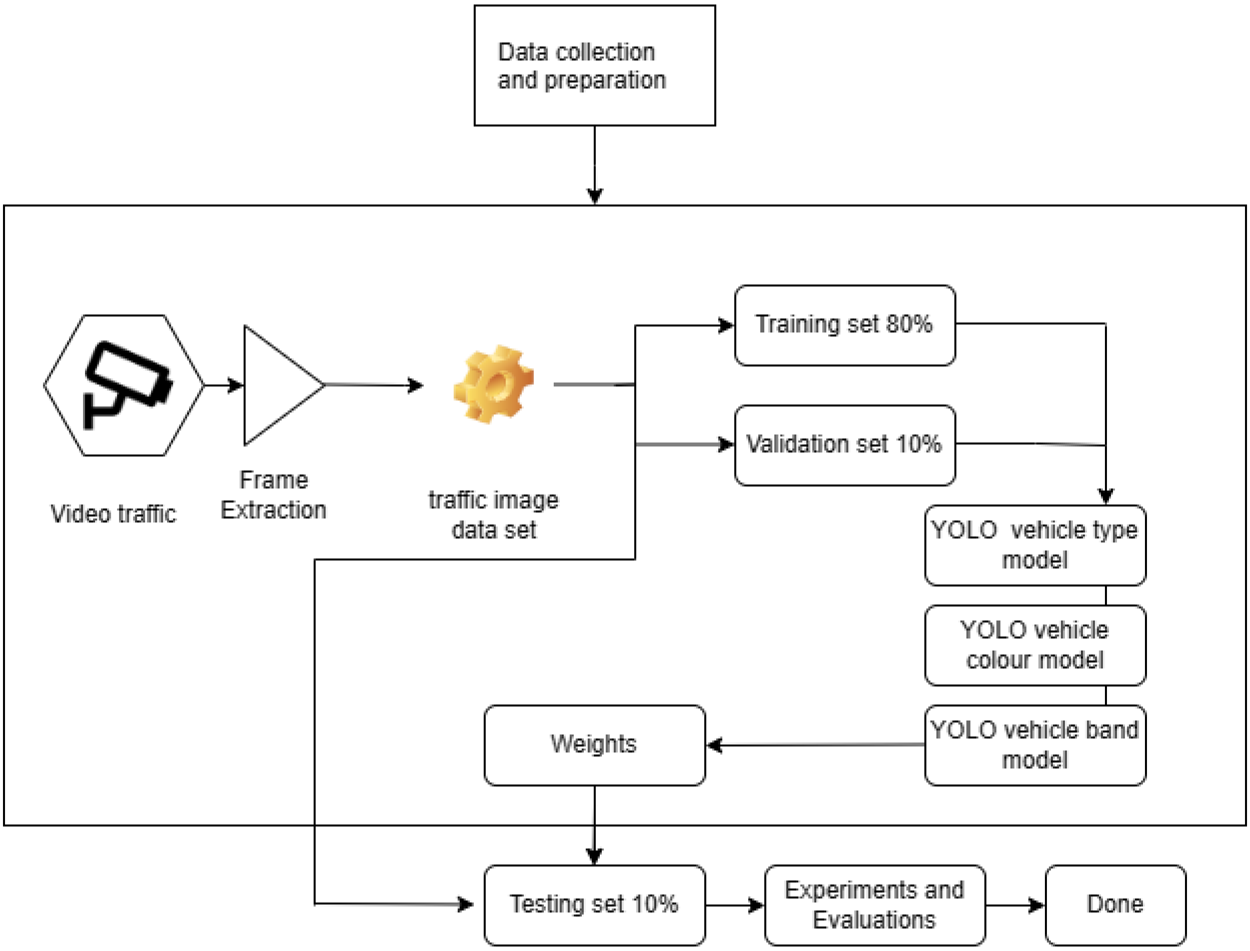


Fig. 1 The research methodology

Figure 1 presents a schematic representation. This visual depiction serves to enhance understanding of the methodology and the overall flow of the research process.

In summary, the research methodology encompasses the collection of video data from CCTV cameras, frame extraction, and the development of deep learning models to identify different attributes of a vehicle's physical appearance, such as its type, colour, and brand.

The research methodology section outlines the approach employed in conducting this research project, which aims to establish a framework for generating deep-learning models to identify various aspects of a vehicle's physical appearance. To facilitate clear communication of the proposed framework, a schematic representation has been created and is presented in Figure 1.

The research methodology involves several steps in generating deep-learning models for vehicle physical appearance identification. The process begins with data collection using CCTV cameras installed at an intersection road in Phuket. The collected data is in the form of video footage, which undergoes frame extraction to convert it into a sequence of images.

Three deep learning models were developed to identify different aspects of the vehicle's physical appearance, namely: (i) vehicle type detection, (ii) vehicle colour detection, and (iii) vehicle brand detection. An optional vehicle model detection is also considered.

The research focuses on four vehicle types: Car, Bus, Pickup truck, and Motorcycle.

Additionally, ten vehicle colours are taken into account, including white, black, grey/silver/bronze, blue, red, yellow, green, blue, orange, and others.

The vehicle brands considered in the study include Toyota, Isuzu, Ford, Mazda, BMW, Mitsubishi, Honda, Suzuki, Nissan, and others.

In conclusion, the research methodology entails the collection of video data, frame extraction, and the generation of deep learning models to identify various aspects of a vehicle's physical appearance, encompassing its type, colour, and brand.

4. YOLO Environment Configuration

The specifications of the computer configuration are shown in Table 1.

Hardware	Parameter
OS	Windows 11 Pro
CPU	Intel Core i5-11400H
GPU	NVIDIA GeForce RTX 3060
SSD	Solidigm SSD P44 Pro
RAM	64 GB

Table 1 The specifications of the computer configuration required for the execution of the YOLO V3-V8 algorithm

The configuration of the software can be described as follows.

PyTorch is an open-source machine learning framework primarily developed by Facebook's AI Research Lab (FAIR). It provides a flexible and dynamic approach to building and training neural networks. PyTorch is widely used in the research and development of various machine learning and deep learning applications. Use with YOLO V4-V8.

Ultralytics is a software company that specializes in creating tools and software libraries for computer vision and deep learning tasks. They are known for developing the popular YOLO V5 framework, which is an extension of the YOLO (You Only Look Once) object detection algorithm. YOLO V5 is designed to be faster and more accurate than its predecessors and has gained popularity in the fields of computer vision and deep learning.

Darknet is an open-source neural network framework that is primarily used for building and training deep neural networks, especially in the field of computer vision. It was created by Joseph Redmon, and it gained significant popularity for its implementation of the YOLO (You Only Look Once) family of object detection algorithms. Darknet's YOLO implementations (such as YOLO V2, YOLO V3, etc.) were some of the first to achieve real-time object detection with impressive accuracy. However, as the field of deep learning has evolved, other frameworks like TensorFlow and PyTorch have gained more popularity due to their flexibility, ease of use, and active development communities.

5. Data Collection

The dataset utilized in this study was gathered from CCTV cameras situated in Ko Kaew Subdistrict, Thep Krasattri Road, Mueang District, Phuket Province. The camera's specifications were the Dahua IPC-HFW5541E-ZE model with 5 MP, IR, Vari-Focal Bullet, WizMind Network Camera, and IP67. The duration of the video footage acquired was 30 minutes, with a specified recording date of March 1, 2022. he first step in video data processing involved the extraction of frames, which facilitated the conversion of detailed videos into individual images. These images were then utilized to generate a training dataset for model development and analysis. he geographical location of the cameras are depicted in Figure 2. Figure 3 illustrates an example of the obtained data.

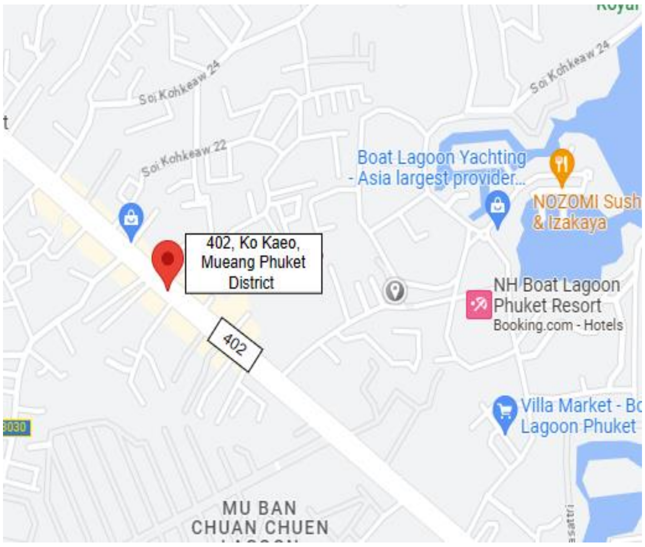

Soi Kohkeaw 24
Soi Kohkeaw 24
Soi Kohkeaw 22
Boat Lagoon Yachting - Asia largest provider...
NOZOMI Sush & Izakaya
402, Ko Kaeo, Mueang Phuket District
NH Boat Lagoon Phuket Resort
Booking.com - Hotels
402
3030
Villa Market - Bo Lagoon Phuket
MU BAN CHUAN CHUEN

Fig. 2 Map showing camera locations Ko Kaeo sub district, Thep krasattri Road, Mueang district, Phuket Province

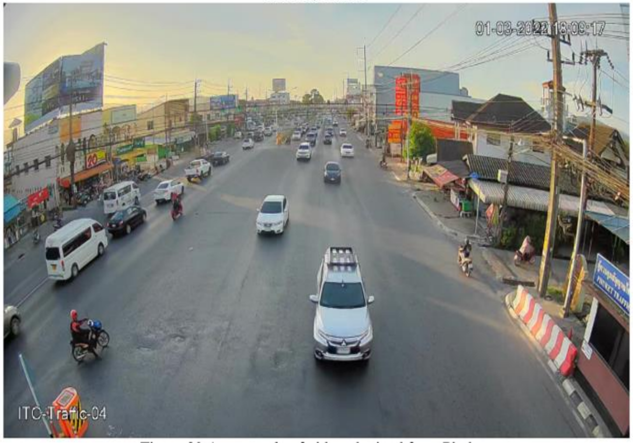


Fig. 3 An example of video obtained from Phuket Province

6. Data Pre-processing

After the completion of data collection, the subsequent step involves preparing the dataset to facilitate the training process of the models. In this research, two essential processes were carried out as part of data pre-processing: frame detection and object labelling. These processes are further elaborated in the following sub-sections.

6.1. Frame Extraction

In the initial phase of data collection, a set of video data was obtained. The first step in processing this data involved converting the videos into a sequence of individual image frames. For instance, a 1-second video was extracted into 3 frames. The extracted frames were then used as the basis for subsequent analysis.

Type	Video	Time (s)	No of frame used in the experiment
Type car	1	15,989	757
Colour car	1	15,989	3,550
Band car	1	15,989	5,283

Table 2 An overview of the frame extraction process

Table 2 provides an overview of the frame extraction process, indicating the number of frames obtained. In this particular case, a total of 1,800 seconds of video data resulted in the extraction of 15,989 frames. From these frames, a subset was selected for further analysis. Specifically, 757 frames, 5,283 frames, and 3,550 frames (which constituted approximately 20% of the data, mixed with other data) were chosen for subsequent processing.

Algorithm 1: Frame Extraction

```
initialize video reader
    open video file
    get video width and height
    set frame rate to 10 fps
    set video duration to 30 minutes
frame_count = frame_rate * (video_duration * 60)  //
Convert video duration to seconds
frame_interval = 1 / frame_rate
output_directory = "path/to/save/frames"
for i = 1 to frame_count:
   current_time = i * frame_interval
   frame =   extract_frame_from_video(current_time)
   save_frame_as_image(frame, i, output_directory)
close video reader
```

The description of the above algorithm can be explained as the following:
Initialize the video reader: This step prepares the necessary resources and libraries to read the video file.
Open video file: The video file is opened for reading the frames.
Get video width and height: Retrieve the dimensions of the video frame to ensure the extracted frames have the correct size.
Set frame rate to 10 fps: Specify the desired frame rate for frame extraction. In this case, it is set to 10 frames per second.
Set video duration to 30 minutes: Define the duration of the video in minutes.
Calculate the frame count: Multiply the frame rate by the video duration in seconds to determine the total number of frames to be extracted.
Calculate the frame interval: The frame interval represents the time interval between each frame based on the desired frame rate.
Specify the output directory: Choose the directory where the extracted frames will be saved as images.
Loop over the frame count: Iterate from 1 to the total number of frames.
Calculate the current time: Multiply the frame index by the frame interval to determine the time in the video corresponding to the current frame.
Extract a frame from the video: Use the current time to retrieve a frame from the video.
Save the frame as an image: Save the extracted frame as an image, using the frame index as part of the image file name. Specify the output directory where the images will be saved. Close the video reader: Release the resources and close the video file after extracting all the frames.

6.2 Labelling

After the frame extraction process, the next step involved object labelling. Object labelling is crucial for identifying and annotating the relevant objects present in the extracted frames, as these objects will be utilized in subsequent processes and analyses.

In the context of this preliminary study, the task of object labelling was facilitated by employing the Roboflow application. Roboflow served as the chosen labelling tool, providing a platform for efficiently and effectively annotating objects of interest within the images.

By utilizing Roboflow, the researchers were able to label and mark the specific objects within the frames, enabling the subsequent stages of analysis and model training to make use of the labelled data. Object labelling is a fundamental step in the process of training and evaluating object detection models, ensuring accurate identification and localization of objects in the images. An example of labelling process using Roboflow is presented in Figure 4.

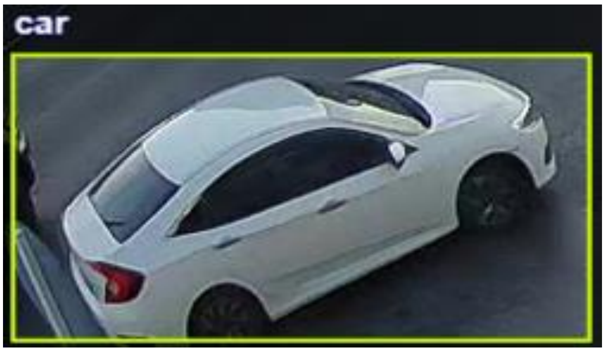


Fig. 4 Labelling using Roboflow

7. Transfer Learning

Transfer learning is a machine learning technique that involves leveraging knowledge gained from solving one problem and applying it to a different but related one. Instead of starting from scratch, transfer learning enables models to benefit from pre-existing knowledge and experience, often obtained from large datasets and complex tasks.

In transfer learning, a model is first trained on a source task, typically involving a large amount of labelled data. The knowledge acquired during this training phase is then transferred or adapted to a target task, which usually has limited labelled data available. By using the knowledge from the source task, the model can generalize better and achieve improved performance on the target task.

In application of Various YOLO Models for Computer Vision-Based Real-Time Pothole Detection [11], the adoption of computer vision techniques for automated pothole detection offers a promising solution to the labor-intensive and time-consuming nature of manual image processing. By utilizing digital imaging and advanced algorithms, can enhance the efficiency, accuracy, and objectivity of road surface monitoring, ultimately facilitating more effective pothole repair and maintenance processes. Fast and Accurate Fish Detection Design with Improved YOLO-v3 model and Transfer Learning [12] In response to the growing demand for monitoring the marine ecosystem, propose an improved version of the You Only Look Once version 3 (YOLOv3) algorithm for real-time fish detection. Our research aims to provide a robust and efficient automated system that benefits researchers in collecting information about marine life. Face mask detection based on transfer learning and PP-YOLO [13] In this paper, the authors propose a face mask detection model called PP-YOLO-Mask, which is built upon the PP-YOLO architecture. The aim of the model is to achieve high accuracy and reasoning speed in face mask detection tasks. To improve the performance of the model, several techniques are employed. First, transfer learning is used, leveraging pre-trained weights from the PP-YOLO model. This allows the model to benefit from the knowledge learned on large-scale datasets and speeds up the training process for face mask detection. Image Recognition of Wind Turbine Blade Defects Using Attention-Based MobileNetv1-YOLOv4 and Transfer Learning [14], in this paper, propose an image recognition method for identifying wind turbine blade defects using attention-based MobileNetv1-YOLOv4 and transfer learning. To reduce complexity and computational requirements, replace the backbone convolutional neural network of YOLOv4 with the lightweight MobileNetv1 for feature extraction. Additionally, introduce attention-based feature refinement through three distinct modules: SENet, ECANet, and CBAM. These modules enable adaptive feature optimization, improving the model's accuracy. In Detection of Unauthorized Unmanned Aerial Vehicles Using YOLOv5 and Transfer Learning [15], detecting drones in surveillance videos poses a significant challenge, primarily due to the difficulty of distinguishing drones from diverse backgrounds. This research paper presents an automated image-based drone detection system that leverages an advanced deep learning-based object detection algorithm called You Only Look Once version 5 (YOLOv5). The proposed system aims to protect restricted territories and special zones by accurately identifying and preventing unauthorized drone incursions. To address the limited availability of training samples in our dataset, transfer learning is employed to pretrain the model and enhance its performance. A novel fine-tuned YOLOv6 transfer learning model for real-time object detection [16] was proposed. The proposed model builds upon YOLOv6 as a baseline model. To improve efficiency in terms of detection accuracy and inference speed, a pruning and fine-tuning algorithm, as well as a transfer learning algorithm, are introduced. These techniques optimize the model's performance while considering computational resources. TransLearn-YOLOx: Improved-YOLO with Transfer Learning for Fast and Accurate Multiclass UAV Detection was proposed by [17]. In summary, this paper introduces deep learning-based solutions using YOLOv5 and YOLOv7 for multi-class UAV classification, aiming to enhance real-time detection accuracy. The utilization of transfer learning facilitates improved performance and accelerated training. The experiments conducted on a customized dataset showcase the positive impact of integrating transfer learning on classification results. Signboard Detection using YOLOv5 with Transfer Learning [18] in summary, this paper presents a deep learning-based solution using YOLOv5 with transfer learning for real-time traffic sign detection. The proposed model demonstrates superior performance compared to other deep learning models, including previous versions of YOLO. The utilization of a dataset comprising various signboard classes contributes to the robustness of the model. The implemented solution showcases high performance and accuracy in traffic sign detection. Application Research of Improved YOLO V3 Algorithm in PCB Electronic Component Detection [19] in summary, this paper presents a deep learning-based solution using YOLOv5 with transfer learning for real-time traffic sign detection. The proposed model demonstrates superior performance compared to other deep learning models, including previous versions of YOLO. The utilization of a dataset comprising various signboard classes contributes to the robustness of the model. The implemented solution showcases high performance and accuracy in traffic sign data this paper contributes to the field of electronic component detection by providing a comprehensive training dataset, introducing an improved YOLO V3 algorithm, and utilizing clustering techniques to optimize anchor box design. These advancements result in more accurate and efficient detection of electronic components in both real and virtual PCB images detection. Leaf-based disease detection in bell pepper plant using YOLO v5 [20] To address this issue, random sampling of images from different parts of the farm is conducted. The YOLOv5 algorithm is employed to detect bacterial spot disease symptoms on the leaves of bell pepper plants. This approach enables the detection of even small disease spots with high speed and accuracy. By inputting random farm pictures captured using a mobile phone, the model swiftly predicts bounding boxes and class probabilities for disease identification to address this issue, random sampling of images from different parts of the farm is conducted. The YOLOv5 algorithm is employed to detect bacterial spot disease symptoms on the leaves of bell pepper plants. This approach enables the detection of even small disease spots with high speed and accuracy. By inputting random farm pictures captured using a mobile phone, the model swiftly predicts bounding boxes and class probabilities for disease identification. Vehicle Detection and Tracking using YOLO and DeepSORT [21] To address this requirement, this paper proposes the implementation of deep learning techniques for vehicle detection. TensorFlow, a machine learning platform, and the You Only Look Once (YOLO) algorithm, an object detection algorithm for real-time applications, are utilized in this project. By combining these tools with Python as the programming language and incorporating other dependencies, the paper demonstrates the enhancement of the latest YOLOv4 algorithm in the vehicle detection system compared to previous models.

In the context of the research presented in this paper, six transfer learning architectures were employed, namely YOLO V3, YOLO V4, YOLO V5, YOLO V6, YOLO V7, and YOLO V8. A concise description of each architecture is presented below. Transfer learning architectures were employed to enhance the performance of object detection models in this study. The following YOLO versions were utilized: YOLO V3, YOLO V4, YOLO V5, YOLO V6, YOLO V7, and YOLO V8. Each architecture represents a specific iteration and improvement upon the YOLO framework. These versions incorporate advancements in deep learning and object detection techniques to achieve higher accuracy and efficiency in detecting and localizing objects in images or videos. By leveraging pre-trained models and transfer learning techniques, these architectures enable the transfer of knowledge from pre-existing models to the specific task of object detection, resulting in improved performance and reduced training time.

7.1. YOLO V3

YOLO V3 is an advanced object detection model that draws inspiration from ResNet and FPN architectures. Darknet-53, the backbone of YOLO V3, consists of 52 convolutional layers and incorporates skip connections (similar to ResNet) and three prediction heads (like FPN). This architecture demonstrates excellent performance across various input resolutions, as evidenced by multiple checkpoints in the GluonCV Zoo with different input resolutions but consistent network parameters. In the COCO-2017 YOLO V3 benchmark, when evaluated at an input resolution of 608x608, the model achieves a competitive mAP score of 37, comparable to the Faster-RCNN-ResNet50 model trained using GluonCV [22].

7.2. YOLO V4

YOLO V4 is an enhanced iteration of YOLO V3, delivering improved performance by employing the CSPDarkNet53 [23] core architecture (CSP stands for Cross Stage Partial). It also incorporates additional features such as Spatial Pyramid Pooling (SPP) and Path Aggregation Network (PAN) [24]. These innovations contribute to increased accuracy and detection speed, positioning YOLO V4 as one of the most efficient object detection models available.

7.3. YOLO V5

YOLO V5 is a version of the popular object detection model developed by Glenn Jocher, implemented using PyTorch. It builds upon the core architecture of YOLO V3, including the CSP backbone, PANet [25], and mosaic data augmentation. While lacking official documentation, YOLO V5 has demonstrated improved performance over its predecessor, although its accuracy may be slightly compromised. However, YOLO V5 boasts significantly faster speed, making it well-suited for real-time applications on CPUs. The CSPNet used in YOLO V5 partitions the input data into two equal parts. One part remains unchanged and undergoes processing through the "change block," while the other is processed through dense and change blocks. This CSPNet connection enables the model to retain crucial information from previous layers while reducing computational complexity [26].

7.4. YOLO V6

YOLO V6 is an advanced single-stage object detection framework designed specifically for industrial applications, prioritizing hardware-friendly efficiency and high performance. It has demonstrated superior accuracy and inference speed compared to YOLO V5, making it the optimal version of the YOLO architecture for production-level applications. YOLO V6 introduces the EfficientRep backbone and Rep-PAN Neck [27], designed with hardware constraints in mind, to improve upon the backbone and neck of YOLO. These innovations enable the model to deliver better performance while remaining computationally efficient, making it a powerful tool for real-world object detection applications [28].

7.5. YOLO V7

YOLO V7 is an object detection architecture that builds upon the foundations of previous models, such as YOLO V4, Scaled YOLO V4, and YOLO-R [29], while introducing innovative enhancements. The architecture has undergone iterative refinement through a series of experiments to improve accuracy and speed. YOLO V7 incorporates the E-ELAN computational block in its backbone, drawing inspiration from research on network efficiency. This block is designed to focus on factors affecting both speed and accuracy, such as model scaling for concatenation-based models and bat-of-freebies features that optimize the network structure and loss function. Leveraging these innovations, YOLO V7 achieves significant improvements in detection speed and accuracy compared to previous YOLO versions [30].

7.6. YOLO V8

YOLO V8 represents a streamlined parameter configuration derived from the YOLO V8 algorithm. This model is composed of three main components consisting of a backbone network, a neck network, and a prediction output head.

The backbone network is centered around convolutional operations, enabling the extraction of diverse-scale features from RGB (Red, Green, Blue) color images. Simultaneously, the neck network plays a crucial role in amalgamating the features obtained from the backbone network. To achieve this, a feature pyramid structure known as Feature Pyramid Networks (FPN) is commonly employed. This structure effectively aggregates low-level features into higher-level representations.

The responsibility of the head layer pertains to predicting the target categories. To facilitate the selection and detection of image contents, three sets of detection detectors with varying sizes are employed. This multifaceted approach enhances the model's ability to accurately identify and classify objects within the images. [31]

8. Experiments and Evaluations

With respect to the work presented in this paper, a series of experiments have been conducted, an overview of which is presented in this section. The initial step involved completing the object labelling process, followed by employing several existing CNN architectures to generate models. Specifically, we focused on six CNN architectures including YOLO V3, YOLO V4, YOLO V5, YOLO V6, YOLO V7, and YOLO V8.

Experimental, first of all, to examine the effectiveness of each algorithm improvement step and the overall performance of the proposed algorithm. To gather the necessary data, we collected it through a CCTV camera located in Phuket. Subsequently, a training dataset was created, and data augmentation techniques were implemented to enhance the dataset's quality and diversity. The data was then extracted and labelled using Roboflow. These labelled datasets were utilized to train the models, employing three of the aforementioned CNN architectures. This document trains and tests each step of improvement on three datasets: (i) vehicle type, (ii) vehicle color type, and (iii) vehicle brand. The vehicle type dataset is 757 images, the vehicle color type is 3,550 images, and the vehicle brand is 5,283 images. The data enhancement technique is used to create differences in the data set to increase the quality to 2,271, 10,650, and 15,849 images arranged according to the data set, each of which is divided into 3 sets, namely the training data set, validation set, and test data set. The dataset for the vehicle type consists of 1,817 training images, 227 validation images, and 227 testing images. The dataset for vehicle color type consists of 8,520 training images. 1,065 validation images, and 1,065 testing images. Finally, the vehicle brand detection dataset consists of 12,679 training images, 1,587 validation images, and 1,583 testing images, which were used to complete the preparatory image and combine with all 3 data enrichment processes. Designs include (i) a 50% probability of horizontal flip, (ii) random rotation between -15 and +15 degrees, and (iii) random shear between -15° and +15° horizontally, and -15° to +15°, then trained and tested each step of the improvement on the dataset and compared with each model of the YOLO mAP in Google Colab. Using the same settings. For more comprehensive details on each experiment set, please refer to Sub-section 8.1, 8.2, and 8.3, respectively.

8.1. Vehicle Type Detection

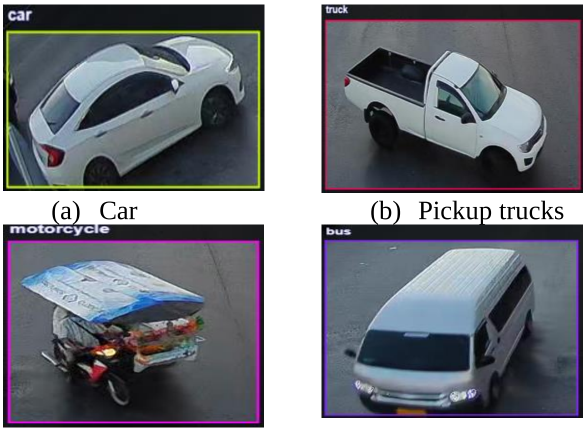
The vehicle type detection model incorporates data augmentation techniques to enhance the dataset, resulting in a total of 2,271 images. These images were divided into four vehicle categories consisting of 687 images of vehicles, 492 images of pickup trucks, 791 images of motorcycles, and 301 images of passenger vehicles. The images from all categories were divided into three sets with 1,817 training set, 227 validation set, and 227 testing set. Examples of vehicle types are presented in Figure 5.

The experiment was conducted over a total of 100 epochs, encompassing all four classes. The batch size was set to 32, and the image size used was 416 x 416. The performance of classification was evaluated using YOLO V3, YOLO V4, YOLO V5, YOLO V6, YOLO V7, and YOLO V8 on the training dataset.

For the training and implementation of YOLO V3 and YOLO V4, a custom framework called Darknet was employed. Darknet is specifically designed to facilitate the training and deployment of YOLO models.

YOLO V5 utilizes a training algorithm known as EfficientNet. EfficientNet is a cutting-edge object detection algorithm that focuses on striking a balance between accuracy and efficiency.

YOLO V6 is similar to YOLO V5 but adopts a variant of the EfficientNet architecture called EfficientNet-L2. This architecture is more efficient compared to the one used in YOLO V5, enhancing overall performance.



(a) Car (b) Pickup trucks

(c) Motorcycles (d) Van

Fig. 5 Vehicle type example images

The most recent iteration of the YOLO algorithm, YOLO V7, introduces several noteworthy enhancements compared to its predecessors. Notably, a significant improvement lies in the incorporation of anchor boxes. Anchor boxes serve as pre-defined bounding boxes with diverse aspect ratios, enabling the detection of objects with varying shapes. In YOLO V7, nine anchor boxes are utilized, facilitating the detection of a broader spectrum of object sizes and configurations compared to earlier versions. YOLO V8 is designed to accommodate a broader spectrum of object detection and segmentation assignments. Leveraging its extensibility, YOLO V8 incorporates several advancements, including a novel backbone network, an anchor-free detection head, and an innovative loss function. As a result of these enhancements, YOLO V8 attains an even higher level of efficiency.

Table 3 shows the evaluation performance for vehicle type detection, presenting the results obtained from the trained models. Notably, YOLO V8 architecture achieves the highest performance with the accuracy value of 98.9%.

Models	mAP.50 (Train)	mAP.50 (Test)
YOLO V3	81.6%	59.8%
YOLO V4	94.7%	62.2%
YOLO V5	98.1%	69.5.%
YOLO V6	98.3%	68.9%
YOLO V7	98.5%	75.8%
YOLO V8	**98.9%**	**83.5%**

Table 3 Detection performance on various vehicle types

Furthermore, Table 3 illustrates a graph depicting the relationship between the number of epochs and the accuracy (mAP.50) achieved in the vehicle type detection task, specifically in the Training Set. And a comparative of vehicle type detection performance in training set and test set in Figure 6.

According to Figure 6, YOLO V5, YOLO V6, and YOLO V7 exhibit similar training speeds but YOLO V8 outperforms its forerunner, YOLO V7, in terms of speed. Speed is a pivotal factor in real-time object detection, and the YOLO V8 offers swifter detection without compromising accuracy. Moreover, YOLO V8 exhibits enhanced precision, particularly in identifying smaller objects, surpassing the capabilities of YOLO V7. The test results indicate that the YOLO V8 achieves a similar mAP.50 for detecting vehicle types as YOLO V5, YOLO V6, and YOLO V7. The graph shows that the YOLO V8 maintains more stability during training.

Each of the six models converges to the same point, with the only distinction being the mAP.50 value. As we move forward, expanding the dataset can facilitate testing additional models, revealing the unique differentiators of each model. Notably, YOLO V3 exhibits lower accuracy compared to other architectures due to its earlier development.

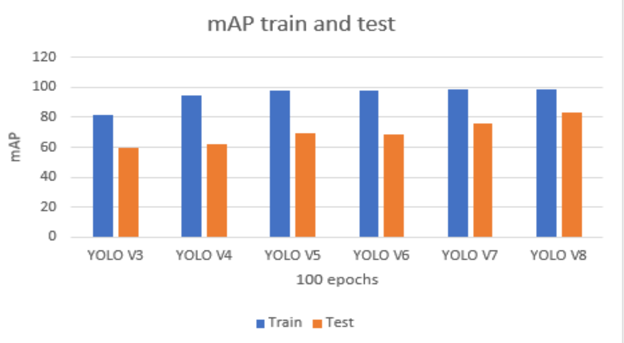
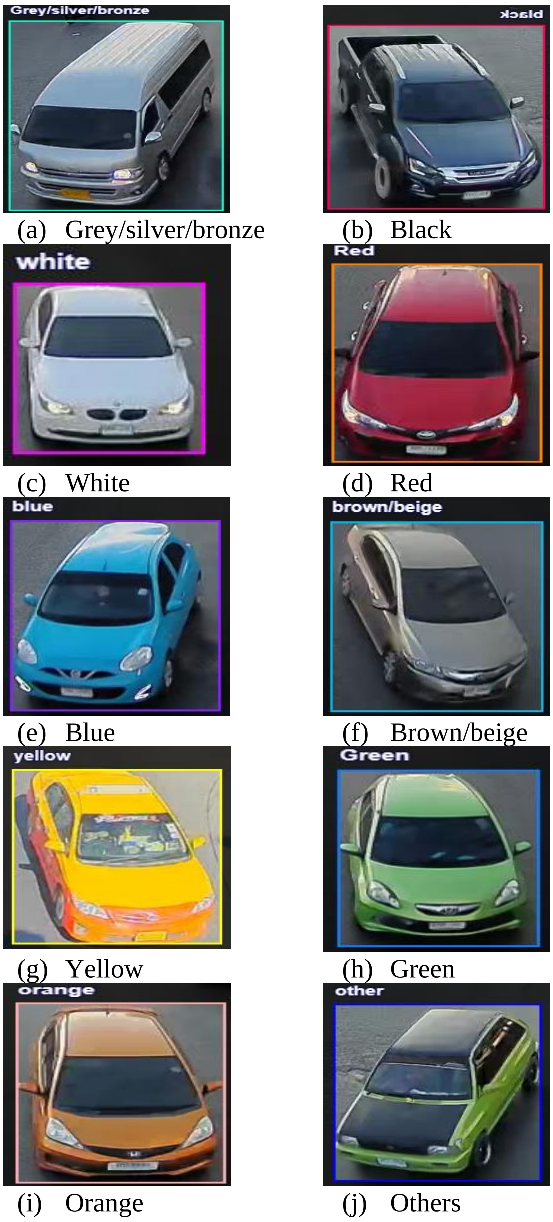


Fig. 6 A comparative of vehicle type detection performance in training set and testing set

Significant improvements in core architecture have been made from YOLO V4 to YOLO V8. Among the vehicle classes, the motorcycle is detected with the highest accuracy, thanks to its distinctive features. Additionally, the YOLO V8 outperforms other YOLO models in accurately capturing the object frame within the device coverage. When applied to the test set, the results are considered reasonable. However, the researcher believes that the dataset's quality is moderate, suggesting that enhancing the dataset's quality would be a viable solution.

8.2. Vehicle Colour Detection

A dataset consisting of 10,650 vehicle colour images was collected for the purpose of vehicle colour detection. To enhance the dataset and ensure accurate labelling, the data was organized into 10 colour categories consisting of 2,160 images of Grey/silver/bronze, 1,960 images of Black, 1,865 images of White, 1,563 images of Red, 1,530 images of Blue, 755 images of Brown/beige, 344 images of Yellow, 263 images of Green, 110 images of Orange, and 100 images of other colours. The sample of each category are shown in Figure 8.



(a) Grey/silver/bronze (b) Black
(c) White (d) Red
(e) Blue (f) Brown/beige
(g) Yellow (h) Green
(i) Orange (j) Others

Fig. 7 Vehicle colour example images

Subsequently, the dataset was divided into three sets to facilitate evaluation comprising of a training dataset with 8,520 images, a validation dataset with 1,065 images, and a testing dataset with 1,065 images.

The experiment was designed with 100 learning epochs, all 10 classes assigned, batch size = 32, and image size = 416 × 416. The classification performance of YOLO V3, YOLO V4, YOLO V5, YOLO V6, YOLO V7, and YOLO V8 was evaluated using the training dataset.

This advancement contributes to a reduction in false positives and enhances the model's accuracy and versatility. The obtained result is presented in Table 4.

Models	mAP.50 (Train)	mAP.50 (Test)
YOLO V3	72.8%	52.4%
YOLO V4	83.7%	60.5%
YOLO V5	87.3%	62.8%
YOLO V6	88.6%	63.3%
YOLO V7	89.2%	70.6%
YOLO V8	**93.5%**	**79.7%**

Table 4 Detection performance on different vehicle colour

The outcomes displayed in Table 4 clearly demonstrate that the YOLO V8 architecture attained an accuracy of 93.5% for vehicle colour detection. Furthermore, Figure 8 provides a graphical representation showcasing the correlation between the number of epochs and the accuracy (mAP.50) achieved during vehicle colour detection in the training set.

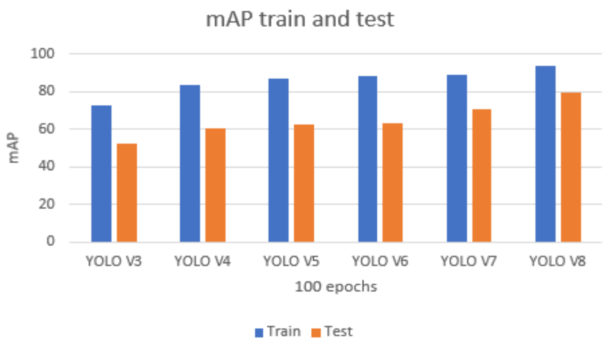
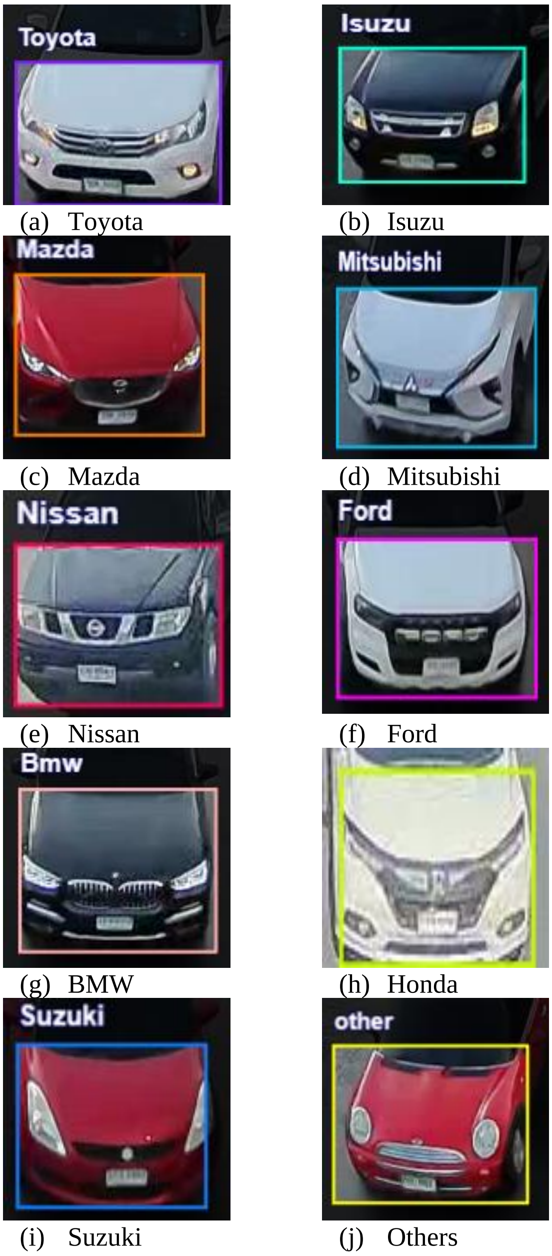


Fig. 8 A comparative of vehicle colour detection performance in training set and testing set

Based on the vehicle colour detection model, Figure 8 demonstrates that the results differ from the previous model due to the additional challenge imposed on the six YOLO models. Notably, the YOLO V8 exhibits a higher mean Average Precision (mAP). To push the limits of development, the six YOLO models were subjected to a vehicle paint test. During this test, difficulties arose in accurately detecting silver, black, and brown vehicle colours, potentially due to their similarity. Addressing this issue may require augmenting the dataset with more relevant data. Regarding colour detection accuracy, red stands out and is detected with the highest precision. Concerning training speed, YOLO V8 is the fastest; however, contrary to the researcher's expectations, YOLO V3 performs better than anticipated in terms of mAP at 50. On the downside, YOLO V3 training speed lags behind the other models, taking two to three times longer. The graph shows that YOLO becomes stable at around 60 epochs or more, while YOLO V7 reaches stability earlier than YOLO V6, YOLO V5, and YOLO V4.

8.3. Vehicle Brand Detection

The dataset used for vehicle brand detection consists of 15,849 images capturing different brands. It includes 10 specific brands, with their respective image counts as follows: (i) Toyota: 2,136 images, (ii) Isuzu: 1,936 images, (iii) Mazda: 1,653 images, (iv) Mitsubishi: 1,728 images, (v) Nissan: 1,763 images, (vi) Ford: 1,693 images, (vii) BMW: 1,469 images, (viii) Honda: 1,854 images, (ix) Suzuki: 1,430 images, and (x) 187 images of other vehicles. The example of each brand is presented in Figure 11.



(a) Toyota
(b) Isuzu
(c) Mazda
(d) Mitsubishi
(e) Nissan
(f) Ford
(g) BMW
(h) Honda
(i) Suzuki
(j) Others

Fig. 9 Vehicle colour example images

The dataset has been partitioned into three distinct sets: a training dataset comprising 12,679 images, a validation dataset containing 1,587 images, and a testing dataset consisting of 1,583 images. The experiment was designed to include 100 epochs, encompassing all ten classes within the dataset. The batch size was set to 32, and the image size used for training and evaluation was 416 x 416. The classification performance was evaluated using YOLO V3, YOLO V4, YOLO V5, YOLO V6, YOLO V7, and YOLO V8 on the training dataset.

Models	mAP.50 (Train)	mAP.50 (Test)
YOLO V3	52.3%	41.3%
YOLO V4	71.5%	52.4%
YOLO V5	80.3%	61.5%
YOLO V6	82.6%	62.3%
YOLO V7	84.6%	69.2%
YOLO V8	**89.8%**	**78.6%**

Table 5 Detection performance on different vehicle brand

According to the findings showcased in Table 5, the YOLO V8 model demonstrated the most exceptional performance, attaining an accuracy of 89.8% for vehicle brand detection. Additionally, Figure 10 illustrates the graphical depiction of the relationship between epochs and accuracy (mAP.50) throughout the training phase for vehicle brand detection on the training set.

Detecting vehicle brands proved to be an initial challenge, albeit less demanding than vehicle colour detection in the previous model. Brand recognition poses a significant challenge for all six YOLO models, as it requires precise and accurate identification of distinct vehicle brands. In this research, the dataset used was of the medium resolution, prompting the researcher to focus on framing the entire hood of the vehicle to enhance the dataset's quality. All six YOLO models were trained to meticulously learn the specific brand details, including vents and logos, in order to achieve optimal training outcomes.

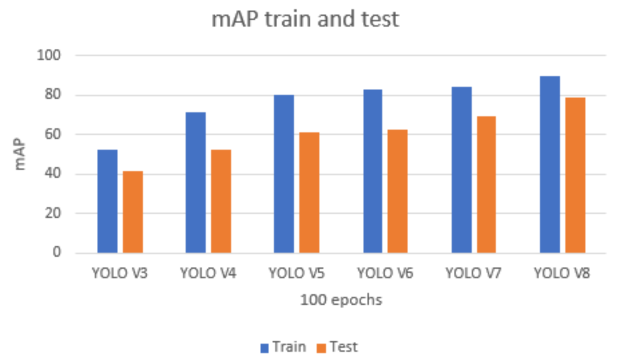


Fig. 10 A comparative of vehicle brand detection performance in training set and testing set

Among the models, the YOLO V8 demonstrated the highest performance, while the YOLO V3 performed comparatively poorly, particularly in detecting small objects. This limitation persists even when using a medium-resolution dataset or framing the entire front of the car. In terms of training time, the YOLO V7 exhibited the fastest training speed among all six models.

During the brand detection process, certain obstacles were encountered. Mitsubishi brand detection proved challenging, while BMW was detected with the highest accuracy. Honda and Suzuki brands were not detected as effectively, potentially due to their similar appearances. To improve accuracy and detection efficiency, additional data should be incorporated into the training process.

To summarize, training the YOLO V8 proved to be the most efficient among the six models. However, challenges persisted in accurately detecting certain vehicle brands, necessitating the inclusion of more data to enhance the accuracy and efficiency of the detection process.

9. Discussion and Conclusion

This research focused on the detection of vehicles, their colours, and brands using surveillance camera data. Transfer learning techniques were employed with the YOLO V3, YOLO V4, YOLO V5, YOLO V6, YOLO V7, and YOLO V8 models.

The results for vehicle detection demonstrated that all six models were able to detect the four types of vehicles to some extent, although there were occasional prediction errors. These errors can be attributed to the limited training data available. Among the models, YOLO V8 exhibited the highest accuracy, achieving a correct prediction rate of 98.9%.

In terms of vehicle colour detection, all six models (YOLO V3, YOLO V4, YOLO V5, YOLO V6, YOLO V7, and YOLO V8) successfully detected the ten different vehicle colours. However, there were instances of mispredictions due to similarities between certain colours. When using YOLO V8 on the entire dataset, the correct prediction rate reached 93.5%.

For vehicle brand classification, YOLO V8 achieved an accuracy rate of 89.8%. It should be noted that the test results were comparatively lower due to the small size of the vehicle symbols. Therefore, optimization methods, such as increasing the volume of data, are necessary to improve performance in this aspect.

In conclusion, the YOLO V8 model demonstrated the highest performance across vehicle detection, colour detection, and brand classification tasks. However, improvements are required to reduce predictive errors caused by limited training data and similarities between colours and vehicle symbols. Increasing the quantity and diversity of data is recommended as an optimization method for future research.

In the future, it would be beneficial to further develop and explore new architectures for YOLO as options for vehicle detection. These advancements could be used by government traffic officers to detect the physical characteristics of vehicles in real-world applications.

Choosing the appropriate image size depends on the specific use case, the characteristics of the dataset, and the available computational resources. If accurate detection of small objects is important, using a larger image size like 640 might be more suitable. On the other hand, if real-time processing is a priority and the loss of some accuracy on small objects can be tolerated, a smaller image size like 416 might be a better choice.

It's worth noting that in practice, experimentation and testing on your specific dataset can help determine the optimal image size that balances accuracy and efficiency

References

[1] WH0 2018. World Health Organization: Road safety, https://https://www.who.int/publication/ i/item/9789241565684/, 2018. [Online; accessed 03-Nov-2021].

[2] T. Ahmad, Y. Ma, M. Yahya, B. Ahmad, S. Nazir, and A. ul Haq, "Object Detection through Modified YOLO Neural Network," Scientific Programming, vol. 2020, pp. 1–10, Jun. 2020, doi: https://doi.org/10.1155/2020/8403262.

[3] Hasan Saribas, Hakan Cevikalp, and Sinem Kahvecioglu, "Car detection in images taken from unmanned aerial vehicles," May 2018, doi: https://doi.org/10.1109/siu.2018.8404201.O. B. R. Strimpel, "Computer graphics," in *McGraw-Hill Encyclopedia of Science and Technology*, 8th ed., Vol. 4. New York: McGraw-Hill, 1997, pp. 279-283.

[4] B. Xu, B. Wang, and Y. Gu, "Vehicle Detection in Aerial Images Using Modified YOLO," Oct. 2019, doi: https://doi.org/10.1109/icct46805.2019.8947049.U. J. Gelinas, Jr., S. G. Sutton, and J. Fedorowicz, *Business Processes and Information Technology*. Cincinnati: South-Western/Thomson Learning, 2004.

[5] Gayatri Sasi Rekha Machiraju, K. Aruna Kumari, and Shaik Khadar Sharif, "Object Detection and Tracking for Community Surveillance using Transfer Learning," Jan. 2021, doi: https://doi.org/10.1109/icict50816.2021.9358698.

[6] Z. Chen, R. Khemmar, B. Decoux, A. Atahouet, and J.-Y. Ertaud, "Real Time Object Detection, Tracking, and Distance and Motion Estimation based on Deep Learning: Application to Smart Mobility," *2019 Eighth International Conference on Emerging Security Technologies (EST)*, Jul. 2019, doi: https://doi.org/10.1109/est.2019.8806222.

[7] A. Corovic, V. Ilic, S. Duric, M. Marijan, and B. Pavkovic, "The Real-Time Detection of Traffic Participants Using YOLO Algorithm," *2018 26th Telecommunications Forum (TELFOR)*, Nov. 2018, doi: https://doi.org/10.1109/telfor.2018.8611986.

[8] X. Zhang, Z. Qiu, P. Huang, J. Hu, and J. Luo, "Application Research of YOLO v2 Combined with Color Identification," Oct. 2018, doi: https://doi.org/10.1109/cyberc.2018.00036

[9] H. S. G. Supreeth and C. M. Patil, "Moving object detection and tracking using deep learning neural network and correlation filter," Apr. 2018, doi: https://doi.org/10.1109/icicct.2018.8473354.

[10] Y. Yang, "Realization of Vehicle Classification System Based on Deep Learning," Jul. 2020, doi: https://doi.org/10.1109/icpics50287.2020.9202376.

[11] S.-S. Park, V.-T. Tran, and D.-E. Lee, "Application of Various YOLO Models for Computer Vision-Based Real-Time Pothole Detection," *Applied Sciences*, vol. 11, no. 23, p. 11229, Nov. 2021, doi: https://doi.org/10.3390/app112311229.

[12] K. Raza and S. Hong, "Fast and Accurate Fish Detection Design with Improved YOLO-v3 Model and Transfer Learning," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 2, 2020, doi: https://doi.org/10.14569/ijacsa.2020.0110202.

[13] W. Jian and L. Lang, "Face mask detection based on Transfer learning and PP-YOLO," *IEEE Xplore*, Mar. 01, 2021. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9389953 (accessed Jan. 20, 2023).

[14] C. Zhang, T. Yang, and J. Yang, "Image Recognition of Wind Turbine Blade Defects Using Attention-Based MobileNetv1-YOLOv4 and Transfer Learning," *Sensors*, vol. 22, no. 16, p. 6009, Aug. 2022, doi: https://doi.org/10.3390/s22166009.

[15] N. Al-Qubaydhi et al., "Detection of Unauthorized Unmanned Aerial Vehicles Using YOLOv5 and Transfer Learning," *Electronics*, vol. 11, no. 17, p. 2669, Aug. 2022, doi: https://doi.org/10.3390/electronics11172669.

[16] C. Gupta, N. S. Gill, P. Gulia, and J. M. Chatterjee, "A novel finetuned YOLOv6 transfer learning model for real-time object detection," *Journal of Real-Time Image Processing*, vol. 20, no. 3, Apr. 2023, doi: https://doi.org/10.1007/s11554-023-01299-3.

[17] Misha Urooj Khan, Mahnoor Dil, Maham Misbah, Farooq Alam Orakzai, Muhammad Zeshan Alam, and K. Chang, "TransLearn-YOLOx: Improved-YOLO with Transfer Learning for Fast and Accurate Multiclass UAV Detection," Jan. 2023, doi: https://doi.org/10.20944/preprints202212.0049.v2.

[18] U. Kulkarni et al., "Signboard Detection using YOLOv5 with Transfer Learning," Apr. 2023, doi: https://doi.org/10.1109/i2ct57861.2023.10126326.

[19] J. Li, J. Gu, Z. Huang, and J. Wen, "Application Research of Improved YOLO V3 Algorithm in PCB Electronic Component Detection," *Applied Sciences*, vol. 9, no. 18, p. 3750, Sep. 2019, doi: https://doi.org/10.3390/app9183750.

[20] M. P. Mathew and T. Y. Mahesh, "Leaf-based disease detection in bell pepper plant using YOLO v5," *Signal, Image and Video Processing*, Sep. 2021, doi: https://doi.org/10.1007/s11760-021-02024-y.

[21] M. A. Bin Zuraimi and F. H. Kamaru Zaman, "Vehicle Detection and Tracking using YOLO and DeepSORT*," IEEE Xplore*, Apr. 01, 2021. https://ieeexplore.ieee.org/abstract/document/9431784 (accessed Nov. 26, 2022).

[22] Z. Yi, S. Yongliang, and Z. Jun, "An improved tiny-yolov3 pedestrian detection algorithm," *Optik*, vol. 183, pp. 17–23, Apr. 2019, doi: https://doi.org/10.1016/j.ijleo.2019.02.038.

[23] Chaima Gouider and Hassene Seddik, "YOLOv4 enhancement with efficient channel recalibration approach in CSPdarknet53," Jul. 2022, doi: https://doi.org/10.1109/itsis56166.2022.10118431.

[24] Y. Cai et al., "YOLOv4-5D: An Effective and Efficient Object Detector for Autonomous Driving," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–13, 2021, doi: https://doi.org/10.1109/tim.2021.3065438.

[25] Z. Xue, H. Lin, and F. Wang, "A Small Target Forest Fire Detection Model Based on YOLOv5 Improvement," *Forests*, vol. 13, no. 8, p. 1332, Aug. 2022, doi: https://doi.org/10.3390/f13081332.

[26] Z. Qiu, Z. Zhao, S. Chen, J. Zeng, Y. Huang, and B. Xiang, "Application of an Improved YOLOv5 Algorithm in Real-Time Detection of Foreign Objects by Ground Penetrating Radar," *Remote Sensing*, vol. 14, no. 8, p. 1895, Apr. 2022, doi: https://doi.org/10.3390/rs14081895.

[27] C. Ji, G. Liu, and D. Zhao, "Ets-3d: An Efficient Two-Stage Framework for Stereo 3d Object Detection," *SSRN Electronic Journal*, 2022, doi: https://doi.org/10.2139/ssrn.4045934.

[28] C. Gupta, N. S. Gill, P. Gulia, and J. M. Chatterjee, "A novel finetuned YOLOv6 transfer learning model for real-time object detection," *Journal of Real-Time Image Processing*, vol. 20, no. 3, Apr. 2023, doi: https://doi.org/10.1007/s11554-023-01299-3.

[29] W. Lan, J. Dang, Y. Wang, and S. Wang, “Pedestrian Detection Based on YOLO Network Model,” IEEE Xplore, 2018. https://ieeexplore.ieee.org/abstract/document/8484698 (accessed Apr. 10, 2021).

[30] B. Xu, B. Wang, and Y. Gu, “Vehicle Detection in Aerial Images Using Modified YOLO,” Oct. 2019, doi: https://doi.org/10.1109/icct46805.2019.8947049.

[31] G. Yang, J. Wang, Z. Nie, H. Yang, and S. Yu, “A Lightweight YOLOv8 Tomato Detection Algorithm Combining Feature Enhancement and Attention,” Agronomy, vol. 13, no. 7, pp. 1824–1824, Jul. 2023, doi: https://doi.org/10.3390/agronomy13071824

Biographies

Kahabodee Prakobchat, born on February 27, 1998, obtained his Bachelor of Science in Technology Petroleum from the Rajamangala University of Technology Srivijaya. Currently, he is pursuing a Master of Science in Data Science at Prince of Songkla University. His research pursuits primarily focus on machine learning techniques, deep learning, and computer vision.

Kwankamon Dittakan holds a PhD in Computer Science from the University of Liverpool, UK. Her expertise lies in artificial intelligence, data science, and machine learning, particularly in analyzing unstructured data such as images, videos, texts, or signals. Currently, she serves as a faculty member at the College of Computing, Prince of Songkla University, Phuket Campus, and leads the Artificial Intelligence Innovation Laboratory (AiiLAB).

Salang Musikasuwan is currently an Assistant Professor in Computer Science at Faculty of Science and Technology, Prince of Songkla University. He received his PhD in Computer Science from The University of Nottingham, United Kingdom. His research interests are machine learning techniques, artificial intelligence, data analytics, and fuzzy set and system applications.